fond_page

Sarant Clémentin

**Delalande Mathieu**

Rapport projet LO07 : Site de covoiturage

Résumé :

*Nous avons réalisé un site web dynamique de covoiturage en utilisant les différentes technologies apprises au cours du semestre.*

Professeur : Lemercier Marc

Semestre : *Printemps 2015*

Table des matières

[Introduction 4](#__RefHeading__206_1738593897)

[I – Analyse du cahier des charges 5](#__RefHeading__208_1738593897)

[II –Modélisation des fonctionnalités 6](#__RefHeading__210_1738593897)

[III – Etapes de conception 7](#__RefHeading__212_1738593897)

[IV – Conception de la base de données 8](#__RefHeading__214_1738593897)

[Conclusion 9](#__RefHeading__216_1738593897)

Introduction

Durant ce semestre nous avons réalisé un site web dynamique de covoiturage. Pour cela nous avons utilisé différents logiciels et langages de programmation web afin de répondre aux exigences du cahier des charges et de proposer un site complet et fonctionnel.

Le site permet aux utilisateurs de rechercher, de proposer et de réserver des trajets après s’être préalablement connecté avec leur login et mot de passe. Seuls les utilisateurs ayant un véhicule ont la possibilité de proposer des trajets. Le site dispose d’un système d’appréciations et d’une messagerie interne permettant aux utilisateurs d’échanger des messages. L’argent des utilisateurs est géré au sein du site, ainsi un conducteur ne recevra l’argent de ses passagers qu’une fois le trajet validé par ce dernier. Il existe une partie administrateur qui permet de visualiser la liste des comptes des personnes inscrites sur le site, et de visualiser la liste des trajets non effectués avec les informations sur le conducteur et les passagers.

Afin de faire face aux différentes contraintes du projet, nous avons utilisé les langages HTML, PHP, CSS, SQL, javascript et utilisé les logiciels NetBeans, MySQLWorkbench, GIT, et la suite WAMP.

Nous avons choisi d’appeler notre site BassBassCar en raison de notre goût prononcé pour la musique.

Nous détaillerons d’abord notre analyse du cahier des charges, ensuite nous expliquerons le fonctionnement de notre site, puis nous nous intéresserons aux étapes de conceptions de notre projet et de la base de données. Enfin, nous conclurons sur les apports que nous a procuré ce projet et sur les points pouvant être améliorés.

I – Analyse du cahier des charges

Au début du projet nous avons pris connaissance du sujet et de son cahier des charges. Afin de rendre un travail le plus abouti possible, nous avons relevé les points importants et les contraintes du cahier des charges.

Nous avons ensuite listé ces points importants en essayant de leurs donner des priorités :

* Un utilisateur ne peut pas accéder au site s’il n’est pas inscrit et connecté.
* Pour s'inscrire, un utilisateur doit rentrer son nom, son prénom, son année de naissance, son login et son mot de passe. Il peut ensuite ajouter une photo de profil ainsi que les informations relatives à son véhicules en se rendant sur son profil.
* Un utilisateur ne peut proposer un trajet que s’il a un véhicule et que celui-ci est assimilé à son profil.
* Un utilisateur peut publier une annonce contenant la date et l’heure de départ, la ville de départ et d’arrivée, le prix et le nombre de places disponibles.
* Un utilisateur peut rechercher un trajet si celui-ci est déclaré non effectué, les villes proposées doivent être gérées dynamiquement.
* Un utilisateur peut réserver un trajet si le nombre de places disponibles est supérieur ou égal à 1.
* Un conducteur peut supprimer son trajet mais les passagers doivent être informés via le système de messagerie interne. Le conducteur doit aussi leur verser une pénalité de 10€.
* Un conducteur peut afficher la liste de ses propres trajets avec la liste des passagers inscrits.
* Le conducteur reçoit l’argent qui lui est dû, une fois qu’il a validé le voyage.
* Chaque passager et chaque conducteur à la possibilité d’évaluer le trajet qu'il a effectué.

II –Modélisation des fonctionnalités



1. Les formulaires s'enregistrer et se connecter

Lors de sa première visite sur le site, l'utilisateur se voit contraint de s'enregistrer si il souhaite se rendre sur les autres pages du site. Il doit alors renseigner les champs suivants : Nom, Prénom, Année de naissance, Login et Password.

Si l'utilisateur oublie de remplir un des champs, un message d'alerte HTML5 le lui rappelle.

En cliquant sur le bouton de validation « S'inscrire », l'utilisateur est redirigé vers la page control-authentification.php qui a pour rôle d'insérer les informations du membres dans la table « membres » de la base de donnée comme ci dessous.



Si l'utilisateur est déjà inscrit, il devra alors s'identifier via le formulaire si contre.

De la même façon que pour l'inscription, en validant le formulaire, l'utilisateur sera redirigé vers la page control-authentification.php dans le but de vérifier si l'utilisateur existe et que le mot de passe est le bon.

2. La barre de recherche de trajet



La barre de recherche de trajet est l'interface dynamique qui permet à l'utilisateur de trouver une covoiturage après avoir sélectionné une ville de départ, une ville d'arrivée et une date. Seules les villes qui possèdent des trajets sont proposés dans la liste déroulante dans le but de faire gagner du temps à l'utilisateur dans sa recherche.

En cliquant sur le bouton rechercher, l'utilisateur est dirigé vers la page dynamique annonces.php qui a pour rôle d'aller chercher dans la base de donnée les trajets dont la ville de départ, la ville d'arrivée et la date correspondent à sa requête. En fonction du nombre de trajets qui coïncident, la page affiche, ou non, les annonces produites via une fonction PHP.

3. Le formulaire « proposer un trajet »

Comme son nom l'indique, ce formulaire permet à un utilisateur, qui a renseigné un véhicule sur sa page personnelle, de proposer un nouveau trajet. Il devra alors remplir les champs concernant la ville de départ, la ville d'arrivée, la date, l'heure, le nombre de places disponibles dans sa voiture et le prix.

En cliquant sur le bouton « terminer », le conducteur est dirigé vers la page control-proposer.php qui a pour but d'insérer le trajet dans la base donnée comme ci dessous. Ensuite, il sera automatiquement redirigé vers la page vos\_annonces.php qui affichera la liste des annonces que l'utilisateur aura proposé. L'intérêt est double puisqu'il permet de confirmer à l'utilisateur que sa requête a été exécuté avec succès et de lui rappeler les trajets qu'il a déjà posté sur le site.

4. Le profil personnel

La page profil\_user.php est une page propre à chacun. Elle comporte les informations que l'utilisateur a renseigné lors de son inscription et les avis que les autres voyageurs lui ont laissé après un trajet. Mais si l'utilisateur a pris le temps de compléter entièrement son profil, la page peut comporter les informations concernant son véhicules ainsi qu'une photo visible par les autres utilisateur lorsqu'il postera un nouveau trajet.

Chaque utilisateur peut ajouter de l'argent sur son compte BassBass Car. En effet, via le bouton « Ajouter de l'argent », l'utilisateur se verra dirigé vers une page simulant un rechargement son compte par un paiement par carte bancaire (voir ci-dessous).

En cliquant sur le bouton « ajouter », son compte sera recharger du montant renseigné.

5. Vos annonces

La page vos\_annonces.php permet à chaque utilisateur d'avoir une vue sur les trajets qu'il a posté. Chacune des annonces se présente comme ci-contre. On peut y retrouver toutes les informations concernant le trajet, le conducteur et son véhicule.

Le conducteur peut, si il le veut, supprimer le trajet en cliquant sur le bouton en forme de croix. Cette action entraînera, via la page control-supprimer\_trajet.php, une suppression du trajet dans la table « trajets », une suppression des lignes correspondantes dans la table «  membres\_has\_trajets » ainsi qu'un dédommagement automatique de 10€ pour les covoitureurs ayant réservé pour ce trajet.

Cependant, il peut aussi confirmé que le trajet a été effectué en cliquant sur le bouton « trajet effectué ». Cette action entraînera, via la page control-trajet\_effectue.php, une suppression du trajet dans la table « trajets » , une suppression des lignes correspondantes dans la table «  membres\_has\_trajets » et un ajout de ce trajet dans la table « trajets\_effectues ». De plus, les covoitureurs inscrits sur ce trajet ainsi que le conducteur seront inscrits dans la table « membres\_has\_trajets\_effectues » en étant reliés au trajet effectué concerné.

6. Vos réservations

La page vos\_reservations.php permet à chaque utilisateur d'avoir une vue sur les trajets qu'il a réservé. Les annonces sont sous la même forme que sur le page vos\_annonces.php à l'exception qu'il n'y a ni bouton pour supprimer le trajet ni bouton pour indiquer que le trajet a été effectué car le trajet a forcément été posté par un autre utilisateur. Si il reste de la place sur ce trajet, il peut cependant y avoir un bouton « reserver » afin d'avoir la possibilité de réserver une nouvelle place pour un ami non-inscrit sur le site.

7. Vos trajets effectués

8. Vos messages



Sous ses aspects simplistes, la fonctionnalité de messagerie privée interne permet de communiquer entre membres du site de façon efficace. La page list\_pm.php est une boite de réception. C'est sur cette page que l'on peut voir la liste des messages reçus lus ou non-lus. Avant de réserver, l'utilisateur peut, à partir du login, créer un nouveau message.

Le destinataire prendra conscience qu'il a un message non lu par l'indication à côté de « vos messages » dans le menu déroulant du header (voir ci contre).



En cliquant sur le titre du message, l'utilisateur est dirigé vers la page read\_pm.php en transmettant l'ID de la conversation à cette dernière. La page affiche alors le contenu de la conversation avec son auteur en face de chaque message ainsi que la date d’envoi.

On peut alors choisir de répondre au message via le formulaire qui le suit.

III – Etapes de conception

Pour mener à bien notre projet, nous avons utilisé plusieurs langages de programmation et plusieurs logiciels. Cependant, nous avons d’abord commencé à travailler au brouillon, sur papier.

Nous avons dans un premier temps fait un croquis de toutes les pages dont le site avait besoin. Nous les avons ensuite réalisées à l’aide du langage HTML, CSS, PHP et SQL. Notre méthode consistait à construire les pages « en dûr » dans un premier temps, avant des les transformer en fonctions PHP et d'implenter les variables liées à la base de données. Ensuite nous avons décidé de créer des pages appelées «control-nom\_page\_à\_vérifier » qui ont pour but d’assurer la liaison entre les différentes pages visibles par l’utilisateur. Ces pages sont chargées de vérifier les données saisies dans les formulaires, d’effectuer les requêtes SQL appropriées et de rediriger l’utilisateur vers la page adaptée.

Pour réaliser notre projet, nous nous sommes appuyés sur différents logiciels :

* NetBeans et Notepad++ qui ont été nos environnements de développements, ils nous ont permis de coder notre site.
* Git qui nous a permis de travailler en collaboration et de gérer la synchronisation et la fusion de nos fichiers. Git nous a permis de nous focalisé sur le code en lui-même plutôt que sur l’assemblage de nos différents travaux.
* Wamp qui nous a permis de visionner le rendu de notre code php par l’intermédiaire d’Apache qui a fait de notre machine un serveur capable d’interpréter le code php. Wamp nous a aussi aidé à gérer notre base de données avec l’outil phpMyAdmin.

Durant ce projet, nous nous sommes efforcés d’avoir en permanence une version de notre site qui fonctionnait. Nous avons ainsi amélioré en permanence notre site en proposant les améliorations demandées par le cahier des charges.

IV – Conception de la base de données

Pour concevoir notre base de données, nous avons d’abord fait un brouillon au papier en relisant le sujet pour oublier le moins d’informations possible. Nous avons ensuite utilisé le logiciel MySQLWorkbench pour modéliser notre base de données à l’aide d’un schéma. Ce logiciel nous a permis de modéliser chaque table ainsi que leurs relations entre elles. Nous avons ensuite exporté notre modélisation à l’aide d’un code SQL généré par le logiciel vers l’interface phpMyAdmin. Au fur et à mesure que le projet a avancé, nous nous sommes rendu compte que certaines tables et liaisons n’avaient pas d’intérêts, nous les avons donc supprimées, nous avons aussi ajouté des colonnes à d’autres tables.



Schéma de la base de données avec le logiciel MySQLWorkbench

Conclusion